IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Date: January 23, 2008
Robbert C. VAN DER LINDEN, et al.	Confirmation No.: 3722
Serial No.: 10/648,760	Group Art Unit: 2109
Filed: August 25, 2003	Examiner: Kabir U. JAHANGIR

For: METHOD AND SYSTEM FOR STORING STRUCTURED DOCUMENTS STORED IN THEIR NATIVE FORMAT IN A DATABASE

Mail Stop Appeal Briefs – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Dear Sir or Madam:

Appellant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on October 23, 2007.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation of Armonk, New York, by virtue of an assignment from the inventor(s) recorded in the U.S. Patent and Trademark Office on August 25, 2003, at Reel No. 014443 and Frame No. 0255.

II. RELATED APPEALS AND INTERFERENCES

The appeal concerning related U.S. Patent Application Serial No. 10/648,752 may directly affect, be directly affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-2, 4-15, and 17-38 have been rejected. Claims 3 and 16 have been cancelled. Appeal is taken from the rejection of claims 1-2, 4-15, and 17-38.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final Office action dated May 23, 2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention relates to "method and system for storing structured documents in their native format in a database system" (pg. 4, lns. 7-8). "[A]n XML document is parsed and a hierarchical node tree comprising a plurality of nodes is generated from the parsed data. The plurality of nodes is stored in one or more records, which in turn are stored on one or more pages. Each node points to its parent, to its attributes, and to its child-elements. Storing the XML document as a hierarchical node tree comprising a plurality of nodes preserves the document's structure and simplifies navigation up and down the tree. Moreover, by storing the nodes of the tree in standard database records, the existing infrastructure for fixed page buffer management, utilities, and operations, e.g., logging, locking and replication, can be utilized" (pg. 4, ln. 17 to pg. 15, ln. 2).

Independent claim 1 recites a method for storing a structured document (202) in its native format in a database. The method includes receiving a structured document (202). *See, e.g.*, pg. 6, lns. 12-14; figs. 2-3. The method also includes generating a hierarchical node tree (208) comprising a plurality of nodes (508), wherein the node tree (208) represents the structured document (202). *See, e.g.*, pg. 6, lns. 15-17; figs. 2-3. In addition, the method includes storing the plurality of nodes (508) in at least one record (500) in the database, wherein each record (500) comprises a node slot array (506), the

node slot array (506) including a plurality of node slots (507), each node slot (507) including a pointer pointing to one of the plurality of nodes (508) in the hierarchical node tree (208). *See, e.g.*, pg. 6, lns. 21-23; figs. 2-4.

Dependent claim 5, which depends indirectly from claim 1, recites wherein each page (502) comprises a plurality of record slots (505), wherein each record slot (505) includes a pointer pointing to a record (500) stored on the page (502). *See, e.g.*, pg. 8, lns. 8-10; fig. 5.

Dependent claim 8, which depends indirectly from claim 1, recites wherein a child pointer (510) points to a node slot (507) pointing to the child node (508) if the child node (508) is a separate node. *See, e.g.*, pg. 7, lns. 22-23; fig. 4.

Independent claim 14 recites a computer readable medium encoded with a computer program for storing a structured document (202) in its native format in a database. The computer program includes instructions for receiving a structured document (202). *See, e.g.*, pg. 6, lns. 12-14; figs. 2-3. The computer program also includes instructions for generating a hierarchical node tree (208) comprising a plurality of nodes (508), wherein the node tree (208) represents the structured document (202). *See, e.g.*, pg. 6, lns. 15-17; figs. 2-3. In addition, the computer program includes instructions for storing the plurality of nodes (508) in at least one record (500) in the database, wherein each record (500) comprises a node slot array (506), the node slot array (506) including a plurality of node slots (507), each node slot (507) including a pointer pointing to one of the plurality of nodes (508) in the hierarchical node tree (208). *See, e.g.*, pg. 6, lns. 21-23; figs. 2-4.

Dependent claim 18, which depends indirectly from claim 14, recites wherein each page (502) comprises a plurality of record slots (505), wherein each record slot (505) includes a pointer pointing to a record (500) stored on the page (502). *See, e.g.*, pg. 8, lns. 8-10; fig. 5.

Dependent claim 21, which depends indirectly from claim 14, recites wherein a child pointer (510) points to a node slot (507) pointing to the child node (508) if the child node (508) is a separate node. *See, e.g.*, pg. 7, lns. 22-23; fig. 4.

Independent claim 27 recites a system for storing a structured document (202) in its native format in a database. The system includes a computer system (104) coupled to at least one data storage device (106). *See, e.g.*, pg. 5, lns. 4-7; fig. 1. The system also includes a database management system (105) in the computer system (104). *See, e.g.*, pg. 5, lns. 12-13; fig. 1. In addition, the system includes a storage mechanism (200) in the database management system (105) for receiving a structured document (202), generating a hierarchical node tree (208) comprising a plurality of nodes (508), wherein the node tree (208) represents the structured document (202), and storing the plurality of nodes (508) in at least one record (500) in the at least one data storage device, wherein each record (500) comprises a node slot array (506), the node slot array (506) including a plurality of node slots (507), each node slot (507) including a pointer pointing to one of the plurality of nodes (508) in the hierarchical node tree (208). *See, e.g.*, pg. 6, lns. 12-17 and 21-23; figs. 2-4.

Dependent claim 30, which depends indirectly from claim 27, recites wherein each page (502) comprises a plurality of record slots (505), wherein each record slot (505) includes a pointer pointing to a record (500) stored on the page (502). *See, e.g.*, pg. 8, lns. 8-10; fig. 5.

Dependent claim 33, which depends indirectly from claim 27, recites wherein a child pointer (510) points to a node slot (507) pointing to the child node (508) if the child node (508) is a separate node. *See, e.g.*, pg. 7, lns. 22-23; fig. 4.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Appellant requests review as to claims 1-2, 4-7, 10-15, 17-20, 23-32, and 34-37, and their rejection under 35 U.S.C. § 103(a) as being unpatentable over "Efficient storage of XML data" by Kanne et al. (hereinafter "Kanne"), in view of U.S. Patent No. 5,758,361 to van Hoff (hereinafter "Hoff").

2. Appellant requests review as to claims 8-9, 21-22, 33, and 38, and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Kanne, in view of Hoff, and further in view of U.S. Patent No. 5,673,334 to Nichani et al. (hereinafter "Nichani").

VII. ARGUMENTS

1. Claims 1, 14, and 27 Are Patentable Over Kanne in view of Hoff

Claim 1 recites a method for storing a structured document in its native format in a database. The method includes receiving a structured document, generating a hierarchical node tree comprising a plurality of nodes, wherein the node tree represents the structured document, and storing the plurality of nodes in at least one record in the database, wherein each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.

Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

(A) Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree"

Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1.

In final the Office action, the Examiner states:

> Kanne does not disclose that record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.
>
> Hoff discloses array with plurality of slots where each slot includes a pointer corresponds to a node in the tree (see element array contains a plurality of slots which represents the tree of a structured document, in Fig. 2 item # 202).
>
> Kanne and Hoff et al. are analogous art because they are from the same field of endeavor of parsing document into tree.

(May 23, 2007 final Office action, pg. 4).

Although "item # 202" in Hoff cited by the Examiner is an array, "item # 202" in Hoff cannot be construed as the "node slot array" recited in claim 1. Specifically, contrary to the Examiner's assertions, the "document content array 202" in Hoff does NOT include any pointers. Rather, each "content element 204" in "document content array 202" stores content (e.g., a start tag, an end tag, a leaf item, etc.) from an HTML document (*see, e.g.*, FIG. 2 of Hoff).

In addition, Hoff actually teaches against parsing and storing a document in tree format. In particular, Hoff states:

> Conventional HTML document editors represent an HTML document simply as trees in a data structure with starttag, endtag, and leaf items respectively corresponding

> to the starttags, endtags, and leaf contents of the HTML document. As a result, in editing the HTML document, these HTML document editors must traverse not only the starttag and endtag items of the HTML document, but also all of the elements of the leaf items. This unfortunately makes the editing process complex, cumbersome, and slow.

(Col. 1, lns. 27-35 of Hoff).

Hoff also states:

> Unlike in conventional data structures representing HTML documents, the HTML document 125 is represented **linearly** by the data structure 200.

(Col. 3, lns. 63-65 of Hoff).

As seen from the passages above, Hoff specifically teaches against representing a document in a hierarchical tree format. Instead, a document in Hoff is represented in linear format. This is particularly evident from the fact that the term "tree" only appears in the "Background of the Invention" section in Hoff and the term "node" is not used once in Hoff.

Therefore, Hoff does not disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1. Since the Examiner has admitted that Kanne also fails to disclose, teach, or suggest the claim element, even if Hoff were combined with Kanne, the combination would neither teach nor suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1.

(B) Examiner has not established a *prima facie case* of obviousness

To establish a *prima facie case* of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge

generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure (*see, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no prima facie case of obviousness has been established. Therefore, claim 1, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff. Given that claims 14 and 27 each recite elements similar to those of claim 1, those claims, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, for at least the same reasons.

2. Claims 1, 14, and 27 Are Patentable Over Kanne, in view of Hoff, and further in view of Nichani

Claim 1 recites a method for storing a structured document in its native format in a database. The method includes receiving a structured document, generating a hierarchical node tree comprising a plurality of nodes, wherein the node tree represents the structured document, and storing the plurality of nodes in at least one record in the database, wherein each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

(A) Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree"

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1.

As discussed above, Kanne and Hoff do not, alone or in combination disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1.

Nichani relates to "a method and apparatus for performing high integrity, high efficiency machine vision inspection of non-rigid packages without limiting the variety of packages, or type of markings on the packages" (col. 3, lns. 56-59 of Nichani).

In the final Office action, the Examiner states:

> Nichani et al. discloses a child pointer points to a node slot pointing to the child node if the child node is a separate node (see Fig. 7).
> Kanne, Hoff, and Nichani et al. are analogous art because they are from the same field of endeavor of parsing document into tree.

(May 23, 2007 final Office action, pg. 7).

First of all, contrary to the Examiner's assertions, Nichani is completely unrelated to structured documents. In fact, the term "document" is not even used in Nichani.

Second of all, FIG. 7 of Nichani "is an abstraction illustrating a Minimum Spanning Tree effected with a one dimensional linked list" (col. 5, lns. 15-16 of Nichani). As clearly illustrated in FIG. 7, the pointer emanating from each node of the "linked list" only points to another node of the "linked list". Hence, contrary to the Examiner's assertions, none of the pointers in FIG. 7 points to a "node slot" of a "node slot array" that is separate from the "nodes in the hierarchical node tree".

Therefore, Nichani does not disclose, teach, or suggest "each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree", as recited in claim 1. Consequently, even if Nichani were combined with Kanne and Hoff, the combination neither teaches nor suggests the claim element.

(B) Examiner has not established a *prima facie case* of obviousness

To establish a *prima facie case* of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure (*see, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no prima facie case of obviousness has been established. Therefore, claim 1, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani. Given that claims 14 and 27 each

recite elements similar to those of claim 1, those claims, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani, for at least the same reasons.

3. Claims 5, 18, and 30 Are Further Patentable Over Kanne in view of Hoff

Claim 5, which depend indirectly from claim 1, recites wherein each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page.

Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

(A) Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page"

Kanne and Hoff do not, alone or in combination, disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5.

In final the Office action, the Examiner states:

> As per claim 5, 18, and 30, Kanne teaches, a method for storing a structured document in its native format in a database. But does not describe that each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page.
>
> Hoff teaches each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page (see plurality of array slot and pointer array points to content element, in Fig. 2 item 206).

(May 23, 2007 final Office action, pg. 5).

Contrary to the Examiner's assertions, the "pointer array 206" in Hoff does not include pointers that point to "record[s] stored on [a] page", as recited in claim 5. Rather, "pointers 208" in the "pointer

array 206" of Hoff point to the "content elements 204" of "document content array 202" (*see, e.g.*, FIG. 2 of Hoff).

Since the Examiner has already construed the "document content array 202" of Hoff as disclosing the "node slot array" and the "content elements 204" of Hoff as disclosing the "node slots" recited in claim 1, the Examiner cannot now construe the "document content array 202" of Hoff as disclosing the "page" and the "content elements 204" of Hoff as disclosing the "records" recited in claim 5.

In addition, claim 1 recites that "each record comprises a node slot array". Thus, the pointer from "each record slot" should be pointing to something that includes a "node slot array" and not merely to an element in an array.

Therefore, Hoff does not disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5. Since the Examiner has admitted that Kanne also fails to disclose, teach, or suggest the claim element, even if Hoff were combined with Kanne, the combination would neither teach nor suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5.

(B) Examiner has not established a *prima facie case* of obviousness

To establish a *prima facie case* of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or

suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure (*see, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no prima facie case of obviousness has been established. Therefore, claim 5, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff. Given that claims 18 and 30 each recite elements similar to those of claim 5, those claims, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, for at least the same reasons.

4. Claims 5, 18, and 30 Are Further Patentable Over Kanne in view of Hoff, and further in view of Nichani

Claim 5, which depend indirectly from claim 1, recites wherein each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page.

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

(A) Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page"

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5.

As discussed above, Kanne and Hoff do not, alone or in combination disclose, teach, or suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5.

In addition, as discussed above, Nichani is completely unrelated to structured documents. In fact, the terms "document", "page", and "record" are never used in Nichani. Hence, Nichani does not cure the deficiencies of Kanne and Hoff. Consequently, even if Nichani were combined with Kanne and Hoff, the combination would neither teach nor suggest "each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page", as recited in claim 5.

(B) Examiner has not established a *prima facie case* of obviousness

To establish a *prima facie case* of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure (*see, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no prima facie case of obviousness has been established. Therefore, claim 5, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani. Given that claims 18 and 30 each

recite elements similar to those of claim 5, those claims, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani, for at least the same reasons.

5. Claims 8, 21, and 33 Are Further Patentable Over Kanne in view of Hoff, and further in view of Nichani

Claim 8, which depend indirectly from claim 1, recites wherein a child pointer points to a node slot pointing to the child node if the child node is a separate node.

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

(A) Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "a child pointer points to a node slot pointing to the child node if the child node is a separate node"

Kanne, Hoff, and Nichani do not, alone or in combination, disclose, teach, or suggest "a child pointer points to a node slot pointing to the child node if the child node is a separate node", as recited in claim 8.

In the final Office action, the Examiner states:

> As per claim 8, 21, and 33, Kanne and Hoff discloses storing structured document in a database. But does not disclose that a child pointer points to a node slot pointing to the child node if the child node is a separate node.
> Nichani et al. discloses a child pointer points to a node slot pointing to the child node if the child node is a separate node (see Fig. 7).
> Kanne, Hoff, and Nichani et al. are analogous art because they are from the same field of endeavor of parsing document into tree.

(May 23, 2007 final Office action, pg. 7).

As discussed above, the pointers emanating from each node of the "linked list" in FIG. 7 of Nichani only points to another node of the "linked list". Thus, contrary to the Examiner's assertions,

none of the pointers in FIG. 7 of Nichani points to a "node slot" of a "node slot array" that is separate from the "nodes in the hierarchical node tree". As a result, Nichani does not disclose, teach, or suggest "a child pointer points to a node slot pointing to the child node if the child node is a separate node", as recited in claim 8. Hence, even if Nichani were combined with Kanne and Hoff, the combination would neither teach nor suggest the claim element.

(B) Examiner has not established a *prima facie case* of obviousness

To establish a *prima facie case* of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure (*see, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no prima facie case of obviousness has been established. Therefore, claim 8, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani. Given that claims 21 and 33 each recite elements similar to those of claim 8, those claims, and the claims that depend therefrom, are patentable over Kanne, in view of Hoff, and further in view of Nichani, for at least the same reasons.

CONCLUSION

On the basis of the above remarks, Appellant respectfully submits that the final rejection should be reversed.

Respectfully submitted,
SAWYER LAW GROUP LLP

Dated: January 23, 2008

/Erin C. Ming/
Erin C. Ming
Attorney for Applicant
Reg. No. 47,797
(650) 475-1449

APPENDIX OF CLAIMS

1. (Previously Presented) A method for storing a structured document in its native format in a database, the method comprising:

receiving a structured document;

generating a hierarchical node tree comprising a plurality of nodes, wherein the node tree represents the structured document; and

storing the plurality of nodes in at least one record in the database,

wherein each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.

2. (Previously Presented) The method of claim 1, wherein generating the hierarchical node tree further comprises:

parsing the structured document into the plurality of nodes; and

linking each of the plurality of nodes via pointers to form the hierarchical node tree.

3. (Cancelled)

4. (Previously Presented) The method of claim 1, further comprising:

storing the at least one record on at least one page.

5. (Original) The method of claim 4, wherein each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page.

6. (Original) The method of claim 5, wherein each node is identified by an identifier comprising a record slot number corresponding to the record slot pointing to the record in which the node resides and a node slot number corresponding to the note slot pointing to the node.

7. (Original) The method of claim 5, wherein a node comprises a plurality of child pointers if the node has children, wherein each of the plurality of pointers points to a child node.

8. (Original) The method of claim 7, wherein a child pointer points to a node slot pointing to the child node if the child node is a separate node.

9. (Original) The method of claim 8, wherein a node slot in a first record in a first page points to a record slot in a second page and a node slot in a second record if the child node is a separate node stored in the second record on the second page.

10. (Original) The method of claim 7, wherein the node further comprises an in-lined character array.

11. (Original) The method of claim 10, wherein a child pointer describes the child by pointing to the in-lined character array.

12. (Original) The method of claim 7, wherein a child pointer describes the child and its value.

13. (Original) The method of claim 1, wherein the structured document is written in Extensible Markup Language.

14. (Previously Presented) A computer readable medium encoded with a computer program for storing a structured document in its native format in a database, the computer program comprising instructions for:

receiving a structured document;

generating a hierarchical node tree comprising a plurality of nodes, wherein the node tree represents the structured document; and

storing the plurality of nodes in at least one record in the database,

wherein each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.

15. (Previously Presented) The computer readable medium of claim 14, wherein generating the hierarchical node tree further comprises:

parsing the structured document into the plurality of nodes; and

linking each of the plurality of nodes via pointers to form the hierarchical node tree.

16. (Cancelled)

17. (Previously Presented) The computer readable medium of claim 14, further comprising:

storing the at least one record on at least one page.

18. (Original) The computer readable medium of claim 17, wherein each page comprises a plurality of record slots, wherein each record slot includes a pointer pointing to a record stored on the page.

19. (Original) The computer readable medium of claim 18, wherein each node is identified by an identifier comprising a record slot number corresponding to the record slot pointing to the record in which the node resides and a node slot number corresponding to the note slot pointing to the node.

20. (Original) The computer readable medium of claim 18, wherein a node comprises a plurality of child pointers if the node has children, wherein each of the plurality of pointers points to a child node.

21. (Original) The computer readable medium of claim 20, wherein a child pointer points to a node slot pointing to the child node if the child node is a separate node.

22. (Original) The computer readable medium of claim 21, wherein a node slot in a first record in a first page points to a record slot in a second page and a node slot in a second record if the child node is a separate node stored in the second record on the second page.

23. (Original) The computer readable medium of claim 20, wherein the node further comprises an in-lined character array.

24. (Original) The computer readable medium of claim 23, wherein a child pointer describes the child by pointing to the in-lined character array.

25. (Original) The computer readable medium of claim 20, wherein a child pointer describes the child and its value.

26. (Original) The computer readable medium of claim 14, wherein the structured document is written in Extensible Markup Language.

27. (Previously Presented) A system for storing a structured document in its native format in a database, the system comprising:

a computer system coupled to at least one data storage device;

a database management system in the computer system; and

a storage mechanism in the database management system for receiving a structured document, generating a hierarchical node tree comprising a plurality of nodes, wherein the node tree represents the structured document, and storing the plurality of nodes in at least one record in the at least one data storage device,

wherein each record comprises a node slot array, the node slot array including a plurality of node slots, each node slot including a pointer pointing to one of the plurality of nodes in the hierarchical node tree.

28. (Previously Presented) The system of claim 27, wherein the storage mechanism further comprises a parser for parsing the structured document into a plurality of nodes and a node tree generator for linking each of the plurality of nodes via pointers to form the hierarchical node tree.

29. (Previously Presented) The system of claim 27, wherein each record is stored in a page.

30. (Original) The system of claim 29, wherein each page comprises a plurality of record slots, each of which includes a pointer pointing to a record stored on the page.

31. (Original) The system of claim 30, wherein each node is identified by an identifier comprising a record slot number corresponding to the record slot pointing to the record in which the node resides and a node slot number corresponding to the note slot pointing to the node.

32. (Original) The system of claim 29, wherein a node comprises a plurality of child pointers if the node has children, wherein each of the plurality of pointers points to a child node.

33. (Original) The system of claim 32, wherein a child pointer points to a node slot pointing to the child node if the child node is a separate node.

34. (Original) The system of claim 32, wherein the node further comprises an in-lined character array.

35. (Original) The system of claim 34, wherein a child pointer fully describes the child by pointing to the in-lined character array.

36. (Original) The system of claim 27, wherein the structured document is written in Extensible Markup Language.

37. (Previously Presented) The system of claim 32, wherein a child pointer describes the child and its value.

38. (Previously Presented) The system of claim 33, wherein a node slot in a first record in a first page points to a record slot in a second page and a node slot in a second record if the child node is a separate node stored in the second record on the second page.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None